

A Two-Stage Framework with Dual-Branch Network for End-to-End 3D Tooth Landmark Detection

Kaibo Shi, Hairong Jin, and Youyi Zheng(✉)

State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
youyizheng@zju.edu.cn

Abstract. 3D tooth landmark detection is a critical task in digital orthodontics. Several existing methods rely on predicting a shared heatmap, necessitating post-processing steps like clustering, while others predict a fixed number of heatmaps for corresponding landmarks. These methods struggle to handle a variable number of landmarks, resulting in limited robustness. To address these issues, we introduce a two-stage 3D tooth landmark detection framework that integrates TeethGNN for segmentation and an end-to-end neural network (TL-DETR) for landmark detection. Our TL-DETR is designed with two branches for predicting fixed landmarks and a variable number of cusp landmarks on individual teeth. The fixed landmark branch is supervised by the ground truth heatmaps of the fixed landmarks, while the cusp landmark branch provides a preset number of heatmaps and probabilities for potential cusp points. We employ bipartite matching loss to measure the discrepancy between the prediction sets and the ground truth cusp heatmaps, facilitating end-to-end inference. We perform a comprehensive analysis and evaluation on the 3DTeethLand dataset. The experimental results demonstrate that our method significantly outperforms existing baseline approaches, exhibiting both efficiency and robustness. (The code will be released at https://github.com/bibi547/TL-DETR).

Keywords: Tooth Landmark Detection · Tooth Segmentation · Two-stage · End-to-End · Heatmap · Geodesic Distance

1 Introduction

Computer-aided Design (CAD) has become an indispensable tool in modern dentistry, particularly in clinical orthodontics. As a critical task within this field, tooth landmark detection aids clinicians in analyzing dental assessments and formulating orthodontic treatment plans [12]. Traditionally, tooth landmarks

K. Shi and H. Jin—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
Y. Wang et al. (Eds.): MICCAI 2024, LNCS 15571, pp. 203–215, 2025.
https://doi.org/10.1007/978-3-031-88977-6_19

are annotated manually, a process that is not only time-consuming and labor-intensive but also heavily dependent on specialized expertise. Despite advances in integrated semi-automatic and fully automated methods, achieving accurate and efficient tooth landmark detection remains a significant challenge due to the intricate geometric features of tooth surfaces and considerable variations among teeth.

In recent years, several deep learning-based methods have been proposed for detecting 3D tooth landmarks. For example, Wei et al. [14] aggregate multi-scale features from the 3D tooth models and decode them to predict distance maps for each landmark category. However, this method requires extracting the cusp from a shared heatmap using k-means clustering, becoming computationally intensive with high-resolution tooth models due to the time-consuming process of calculating the distance matrix. Additionally, k-means clustering necessitates a preset number of clusters, further limiting the robustness. Wu et al. [16] propose a two-stage framework for 3D tooth segmentation and landmark detection. Their correspondence-based method correlate predicted heatmaps with specific landmarks, eliminating the need for post-processing operations. However, the necessity for a preset number of landmarks limits the model's generalization capabilities. Thus, a significant challenge lies in detecting a variable number of landmarks on 3D tooth models, which renders these methods unavailable for handling challenging cases, such as partially erupted and severely decayed teeth.

To tackle these challenges, we present a two-stage framework, composed of tooth segmentation and landmark detection, for 3D tooth landmark detection. Our framework not only eliminates the need for post-processing but also exhibits excellent generalization capabilities. The overall pipeline of our framework is depicted in Fig. 1. In the first stage, TeethGNN [20] is employed for 3D tooth segmentation, with the results further refined using graph-cut. Subsequently, tooth patches are cropped from the original high-resolution model based on these segmentation outcomes. In the second stage, our tooth landmark detection network, named TL-DETR, is specifically designed to pinpoint landmarks on individual teeth in an end-to-end manner. The architecture begins with the TeethGNN encoder to extract geometric features. This is followed by two decoding branches that respectively predict fixed and cusp landmarks. For the fixed landmarks, TL-DETR directly predicts the corresponding heatmaps. In the cusp landmark branch, TL-DETR predicts both heatmaps and probabilities for potential cusp landmarks. Training for the cusp branch utilizes a bipartite matching loss, optimizing the correspondence by computing a cost matrix between the prediction sets and the ground truth. This approach enables the filtering of 'background' predictions during inference, allowing for a variable number of cusp landmarks that adapt to different tooth shapes.

To evaluate our method, we conduct comprehensive experiments and analyses on the 3DTeethLand dataset extended from Teeth3DS [2] for tooth landmark detection. Both qualitative and quantitative results demonstrate that our TL-DETR achieves efficient and robust performance.

In summary, our contributions are as follows:

- We utilize a two-stage 3D tooth landmark detection framework, which primarily consists of TeethGNN and TL-DETR.
- We introduce a novel end-to-end method for landmark detection on individual teeth. The network is divided into two branches: one predicts fixed landmarks, while the other performs end-to-end prediction of a variable number of cusp landmarks.
- We validated our method on the 3DTeethLand dataset, demonstrating robust performance across various challenging tooth cases and achieving excellent results.

2 Related Works

2.1 3D Tooth Mesh Processing

Geometric processing of 3D shapes is an essential task in computer vision and graphics. With the expansion of deep learning across text and image domains, applications involving 3D point clouds and meshes have garnered extensive research interest. As pioneers in deep point cloud processing, PointNet [10] and PointNet++ [11] introduce feature transformation and hierarchical aggregation of features within radius neighborhoods to capture local features. In contrast, DGCNN [13] leverages k-nearest neighbors in the feature space, enhancing long-distance information transmission. Furthermore, MeshCNN [6] and SubdivNet [7] incorporate mesh topology into deep geometric learning, crafting specialized convolutional and pooling operations tailored for mesh structures. Although these general methodologies offer valuable insights for 3D dental analysis, their direct application is limited by the intricate geometry of tooth surfaces. Therefore, TSGCNet [18] introduces a two-stream graph convolutional network that processes coordinates and normals through distinct branches to enhance the differentiation of cells around tooth boundaries. TSegNet [4] identifies tooth centroids to determine positions and then performs binary segmentation on cropped patches. However, this approach heavily relies on the quality of the cropping. In contrast, TeethGNN [20] predicts face-wise semantic labels and offsets relative to tooth centroids, employing clustering and voting to achieve segmentation results. While this method requires prior knowledge to fine-tune hyperparameters for post-processing, it can achieve high accuracy in handling some challenging cases.

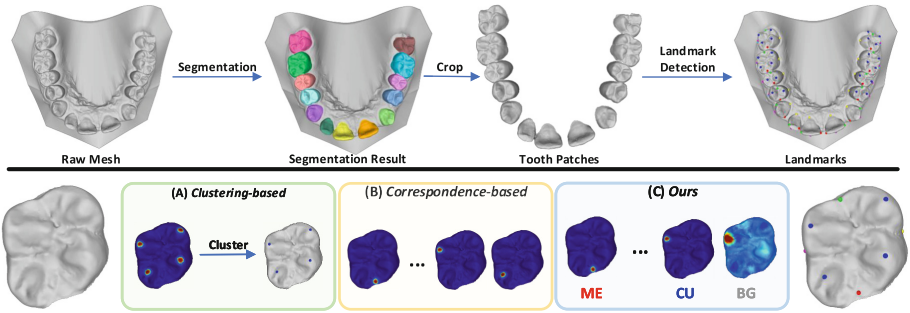


Fig. 1. Top: The overall pipeline of our framework. We first segment the jaw model and then predict landmarks on cropped tooth patches. Red points indicate the Mesial, blue points mark the Cusp, green points represent the Distal, yellow points designate Inner points, cyan points denote Outer points, and pink points signify the Facial Axis points. **Bottom:** Different types of 3D tooth landmark detection methods. The clustering-based method employs clustering algorithms to extract tooth landmarks from a shared heatmap. The correspondence-based method generates a distinct heatmap for each tooth landmark. Our method introduces the 'background' heatmaps to facilitate end-to-end landmark detection. (Color figure online)

2.2 3D Tooth Landmark Detection

While numerous methods excel in 3D tooth segmentation, there is limited research into landmark detection tasks that necessitate fine-grained distinction of local geometric details. Traditional keypoint detection methods rely heavily on hand-crafted features, focusing primarily on local geometric information such as curvature and normals, yet often overlooking semantic information. Although these methods are adept at identifying sharp features in mechanical components, they lack robustness when dealing with complex 3D shape surfaces, such as teeth. Recent advancements in deep learning applied to 2D imaging and 3D volumes have inspired some methods for landmark detection in 3D shapes. KeypointNet [17] introduces a large-scale 3D keypoints dataset, utilizing classification-based methods for saliency and correspondence estimation. However, the sparse distribution of keypoints on 3D shapes leads to sample imbalance issues when simply classifying points. Wei et al. [15] propose a strategy that combines confidence maps with voting, but it still requires post-processing steps such as non-maximum suppression (NMS) and clustering. Wei et al. [14] and Wu et al. [16] respectively predict heatmaps corresponding to categories and instances to achieve 3D tooth landmark detection. Nevertheless, their methods exhibit limited generalization capabilities and struggle with variable numbers of landmarks.

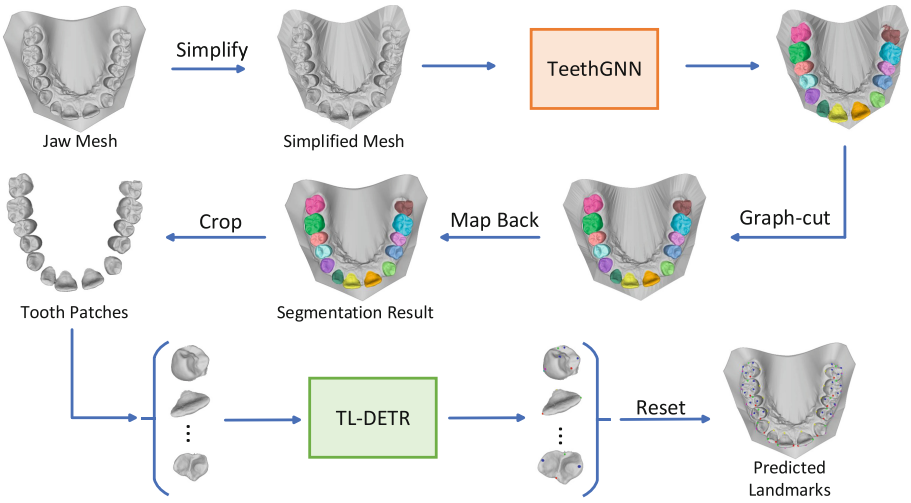


Fig. 2. The detailed pipeline of our framework for 3D tooth landmark detection.

3 Method

In this section, we elaborate on our 3D tooth landmark detection method, as illustrated in Fig. 2. The framework comprises two key stages: 3D tooth segmentation and landmark detection. Initially, we utilize TeethGNN [20] for tooth segmentation, followed by cropping tooth patches from the jaw model according to these segmentation results. Subsequently, our TL-DETR for individual teeth comprises two branches: a fixed landmark branch and a cusp landmark branch. The fixed landmark branch generates a set of heatmaps corresponding to a fixed number of landmarks, while the cusp landmark branch predicts both heatmaps and probabilities for potential cusp landmarks. Furthermore, we employ a bipartite matching loss function to align the prediction sets with the ground truth cusp landmarks, facilitating end-to-end inference for a variable number of cusp landmarks.

3.1 Tooth Segmentation

Since teeth are tightly arranged along the dental arch, accurately predicting tooth landmarks directly from a jaw model is highly challenging. Detecting landmarks on individual teeth can significantly simplify this task. Consequently, 3D tooth segmentation is an essential preliminary step.

The goal of tooth segmentation is to assign a label to each facet of the jaw model, with each model (including both upper and lower jaws) being categorized into 17 classes, comprising 16 teeth and gingiva. The specific process is described as follows: Due to the high resolution of jaw meshes, which can exceed 100,000 facets, direct application in neural network training and inference poses

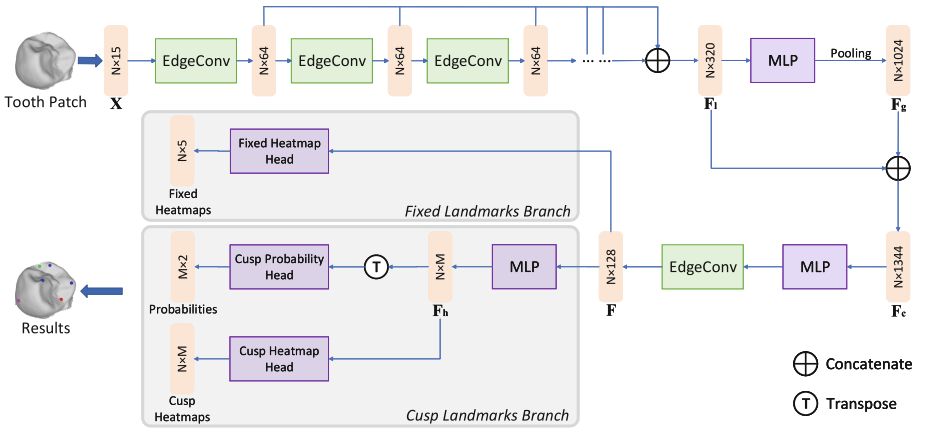


Fig. 3. The network architecture of TL-DETR for individual tooth.

a challenge. Initially, we utilize a mesh decimation algorithm [5] to reduce the original mesh to approximately 10,000 facets. Then, given a simplified mesh with N_s facets, we extract facet features $\mathbf{X_s} \in \mathbb{R}^{N_s \times 15}$ to feed into TeethGNN. The 15-dimensional input features $\mathbf{X_s}$ encompass facet centroid coordinates, facet normals, and vectors from each of the three vertices to the centroid. The semantic labels P_s predicted by TeethGNN [20] are refined using the graph-cut algorithm to enhance the definition at tooth boundaries, yielding refined labels P_r. We then project P_r back onto the original mesh. Each face in the original mesh is assigned the label of its nearest face from the simplified mesh to achieve the final segmentation results P. Ultimately, the individual tooth set with N_t teeth $T = \{t_i\}_{i=1}^{N_t}$ is cropped based on P. Notably, due to the symmetrical distribution of landmarks between the left and right teeth, we merely translate each tooth mesh to the origin for subsequent landmark detection without performing any rotational alignment.

3.2 Tooth Landmark Detection

Data Preparation. The Mesial, Distal, Inner, Outer, and Facial Axis Points are 5 fixed landmarks present on every tooth. However, the number of Cusp Points varies based on the tooth's shape and type. TL-DETR is aimed at predicting heatmaps of these landmarks for individual teeth. To achieve this, we first assign each ground truth landmark from the jaw model to the nearest tooth patch. For an individual tooth t_i with N facets, the corresponding landmark set is $K = \{k_i\}_{i=1}^{N_k}$. We then calculate the geodesic distances between the centroids of all facets $C = \{c_i\}_{i=1}^{N}$ and each landmark k_i, resulting in a geodesic distance matrix. Finally, we apply the Gaussian kernel to the geodesic distance matrix to obtain the ground truth heatmaps $H = \{h_{i,j}\} \in \mathbb{R}^{N \times N_k}$, which can

be formulated as follows:

$$h_{i,j} = \exp^{-\frac{1}{2\sigma^2} d(c_i, k_j)}. \tag{1}$$

Here, $h_{i,j}$ represents the heat value for the i-th facet corresponding to the j-th landmark, d denotes the geodesic distance, and σ is the decay factor. Afterward, H is divided into fixed landmark heatmaps $H_f \in \mathbb{R}^{N\times 5}$ and cusp landmark heatmaps $H_c \in \mathbb{R}^{N\times N_c}$, where N_c is the number of cusp landmarks.

Network Architecture. We introduce an end-to-end tooth landmark detection network, as shown in Fig. 3. Given a single tooth patch t_i with N facets, TL-DETR predicts heatmaps for the fixed landmarks, and both heatmaps and probabilities for the potential cusp landmarks. Specifically, similar to tooth segmentation, we first extract $\mathbf{X} \in \mathbb{R}^{N\times 15}$ facet features as input to the network. We then employ the TeethGNN [20] encoder comprising five static EdgeConv layers, to generate the local geometric features $\mathbf{F_l} \in \mathbb{R}^{N\times 320}$. These features are subsequently passed through fully-connected and pooling layers to derive global features $\mathbf{F_g} \in \mathbb{R}^{N\times 1024}$. The concatenated features, $\mathbf{F_c} = \mathbf{F_l} \oplus \mathbf{F_g}$, are decoded through a static EdgeConv to produce $\mathbf{F} \in \mathbb{R}^{N\times 128}$.

This is followed by two decoding branches for fixed and cusp landmarks, respectively. For the fixed landmark branch, $\mathbf{F}$ is directly fed into a fixed heatmap head composed of an MLP and Conv1d, generating heatmaps $\mathbf{M_h^f} \in \mathbb{R}^{N\times 5}$ corresponding to the 5 fixed landmarks. Simultaneously, the cusp landmark branch decode $\mathbf{F}$ through an MLP and Conv1d into the heatmap feature $\mathbf{F_h} \in \mathbb{R}^{N\times M}$, where M is the preset number of queries. Considering the intrinsic connection between heatmaps and probabilities of landmarks, we input $\mathbf{F_h}$ and the transposed $\mathbf{F_h}$ into heatmap and probability heads respectively to yield predicted heatmaps $\mathbf{M_h^c} \in \mathbb{R}^{N\times M}$ and binary probabilities $\mathbf{M_p} \in \mathbb{R}^{M\times 2}$. The two categories within $\mathbf{M_p}$ denote the cusp landmark and 'background'.

During inference, we select the centroids of facets with the highest values in the predicted heatmaps as the fixed and potential cusp landmarks. Based on $\mathbf{M_p}$, we filter out the potential cusp landmarks predicted as 'background'. Ultimately, we map the predicted landmarks from each tooth patch back onto the jaw model.

Loss Function. According to the two decoding branches, the loss of TL-DETR is also composed of two parts:

$$\mathcal{L} = \lambda_f \mathcal{L}_f + \mathcal{L}_c, \tag{2}$$

where $\mathcal{L}_f$ and $\mathcal{L}_c$ represent the losses on the fixed landmarks and cusp landmarks branches, respectively; λ_f is the hyper-parameter. In the fixed landmarks branch, considering there are corresponding ground truth and predicted heatmaps for each fixed landmark, we use the mean square error (MSE) loss to calculate the heatmap loss. The loss for 5 fixed landmarks is as follows:

$$\mathcal{L}_f = \sum_{i=1}^{5} \left\| h_i - \hat{h_i} \right\|^2, \tag{3}$$

h_i and $\hat{h_i}$ signify respectively the ground truth and predicted heatmaps.

In the cusp landmarks branch, since the number of cusp landmarks is not fixed, we employ the bipartite matching loss inspired by KeypointDETR [8]. First, we construct a cost matrix $\mathcal{M} = \{m_{i,j}\} \in \mathbb{R}^{N_c \times M}$ between the ground truth and predicted cusp landmarks. Considering both heatmaps and probabilities, $\mathcal{M}$ is calculated as shown in the following equation:

$$m_{i,j} = \lambda_h \cdot \sum \left\| h_i - \hat{h_j} \right\|_1 + \lambda_p \cdot (1 - \log \hat{p}_j), \tag{4}$$

where λ_h and λ_p are the hyper-parameters assigned to heatmaps and probabilities to balance the costs. $\hat{p_j}$ signifies the probability of cusp landmark.

Leveraging the Hungarian algorithm [9], we obtain the optimal bipartite matching from $\mathcal{M}$. For positive predictions that match the ground truth cusp landmarks, we use the cross-entropy (CE) loss to compute the probabilities loss and mean square error (MSE) loss to calculate the heatmap loss. As for negative predictions, we assign the corresponding ground truth label as 'background' and only compute the probability loss. We take λ_{ce} and λ_{mse} as coefficients used to balance the loss terms; p_i denotes the ground truth label. The loss of the cusp landmarks branch is calculated as follows:

$$\mathcal{L}_c = \sum_{i=1}^{M} \left(-\lambda_{ce} p_i \log(\hat{p}_i) + \lambda_{mse} \mathbb{1}_{\{p_i \neq 0\}} \frac{1}{N} \sum \left\| h_i - \hat{h_i} \right\|^2 \right). \tag{5}$$

4 Experiment

4.1 Implementation Details

During data preparation, we set σ in Eq. 1 to 0.5. For the cost matrix, λ_h and λ_p are set to 1 and 100, respectively. In the loss function, λ_{ce} and λ_{mse} are set to 1 and 10, respectively. λ_f for the fixed landmark loss is set to 10. The k value within the static EdgeConv is 20 in both TeethGNN [20] and TL-DETR. The number of queries in TL-DETR is set to 50. The maximum number of epochs for training the TeethGNN and TL-DETR networks is 50. Additionally, the learning rate is 0.001 for TeethGNN and 0.0001 for TL-DETR.

4.2 Dataset

For 3D tooth segmentation, we utilize the Teeth3DS dataset [1–3], which comprises 1,800 jaw meshes. From this dataset, 3DTeethLand selects 240 jaw meshes for landmark annotation. We randomly take 30 meshes for the test set, 10 for the validation set, and the remaining for the training set.

Table 1. Quantitative results of different tooth categories with the metrics of mAP and mAR under various distance thresholds.

ϵ	mAP				mAR			
	0.5	1.0.	2.0.	3.0	0.5	1.0.	2.0.	3.0
Incisor	0.446	0.850	0.946	0.952	0.625	0.895	0.970	0.975
Canine	0.314	0.704	0.895	0.909	0.537	0.810	0.937	0.947
Premolar	0.423	0.813	0.945	0.954	0.878	0.793	0.961	0.965
Molar	0.301	0.700	0.923	0.935	0.506	0.842	0.985	0.990

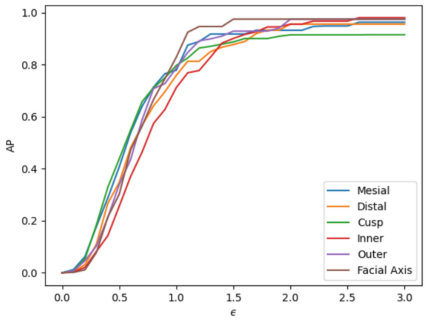


Fig. 4. AP-distance threshold curves for various landmark categories.

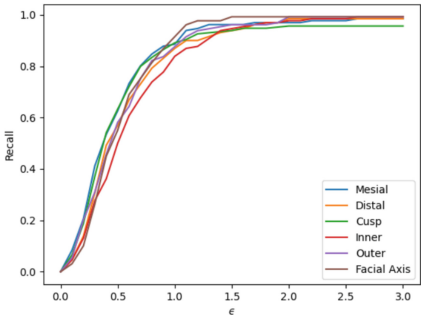


Fig. 5. Recall-distance threshold curves for various landmarks categories.

4.3 Metrics

We evaluate 3D landmark detection performance on jaw models using mean Average Precision (mAP) and mean Average Recall (mAR) at different distance thresholds. These metrics are calculated by averaging the AP and AR across all landmark categories, respectively. Notably, premolars and molars have all six types of landmarks, while incisors and canines have only five, as they lack cusp points. Given distance threshold ϵ, AP represents the area under the Precision-Recall curve, while AR reflects the area under the Recall-Distance threshold curve within the range from $\epsilon - 0.5$ to ϵ. The formulas for Precision (P) and Recall (R) are as follows:

$$P = \frac{TP}{TP + FP}, \tag{6}$$

$$R = \frac{TP}{TP + FN}. \tag{7}$$

Here, TP represents the number of ground truth landmarks that are correctly detected, FP refers to the number of incorrectly predicted landmarks, and FN is the number of ground truth landmarks that are not detected.

Table 2. Quantitative comparison of various tooth segmentation methods in terms of Accuracy (Acc), mean Intersection over Union (mIoU), and Dice Similarity Coefficient (DSC). Lower denotes the mandible, while Upper represents the maxilla.

	Acc(%)		mIoU(%)		DSC(%)	
	Lower	Upper	Lower	Upper	Lower	Upper
DGCNN [13]	89.26	92.94	83.43	87.52	94.09	95.73
MeshSegNet [16]	74.19	83.46	66.90	76.54	86.97	88.30
Point Transformer [19]	85.28	89.71	78.02	84.08	89.97	92.89
TSGCNet [18]	90.27	92.59	85.24	86.37	94.80	95.56
TeethGNN [20]	90.20	95.18	86.00	90.65	95.21	97.09
TeethGNN+Graph-cut	**91.64**	**96.20**	**90.56**	**94.76**	**95.81**	**97.61**

Table 3. Quantitative results of the comparison methods and the variations of our method.

ϵ	mAP				mAR			
	0.5	1.0.	2.0.	3.0	0.5	1.0.	2.0.	3.0
Classification-based	0.300	0.651	0.800	0.812	0.279	0.685	0.849	0.862
Clustering-based	0.283	0.648	0.792	0.810	0.284	0.697	0.855	0.869
DGCNN [13] Encoder	0.345	0.765	0.932	0.949	0.311	0.788	0.965	0.982
TSGCNet [18] Enccoder	0.157	0.372	0.589	0.666	0.291	0.496	0.671	0.729
w/o Graph-cut	0.318	0.697	0.860	0.878	0.301	0.772	0.952	0.970
$\sigma = 0.25$	0.277	0.632	0.877	0.899	0.274	0.702	0.928	0.955
$\sigma = 0.75$	**0.354**	0.765	0.913	0.933	0.304	0.786	0.958	0.972
Ours	0.348	**0.778**	**0.952**	**0.961**	**0.318**	**0.792**	**0.969**	**0.984**

4.4 Results

In Fig. 4 and Fig. 5, we present the AP and recall curves for each landmark category across different distance thresholds. Notably, the detection performance of Cusp landmarks is comparable to that of other fixed landmarks, mainly due to the distinctive geometric features of Cusp landmarks. This also underscores the effectiveness of our cusp landmark branch, trained with bipartite matching loss, in predicting cusp landmarks end-to-end. Consistently, we show the performance of our method in landmark detection across various tooth categories in Table 1. These results further demonstrate that TL-DETR performs equally well on premolars and molars (with cusp landmarks) as on incisors and canines (without cusp landmarks), highlighting its strong generalization and robustness.

Additionally, we compare our TL-DETR with classification-based and clustering-based methods, as shown in the first sub-table of Table 3. In the clustering-based method, we aggregate the cusp landmarks into a shared heatmap and adjust the output of TL-DETR to 6 dimensions: 5 fixed landmark heatmaps and 1 cusp landmark heatmap. Clustering and NMS algo-

rithms are then applied to extract cusp landmarks from the shared heatmap. For the classification-based method, we utilize DGCNN [13] to classify the points directly. We can observe that TL-DETR significantly outperforms these approaches, demonstrating that our end-to-end method requires no post-processing and can accurately predict a variable number of landmarks.

4.5 Ablation Study

To verify the effectiveness of each component in our model, we conduct a series of ablation experiments.

Effects of Tooth Segmentation. The quality of tooth segmentation results plays a crucial role in our method, as high-quality segmentation patches significantly enhance landmark detection accuracy. To this end, we evaluate various tooth segmentation methods on three metrics: Accuracy (Acc), mean Intersection over Union (mIoU), and Dice Similarity Coefficient (DSC). The tooth segmentation experiments are conducted on the Teeth3DS [1–3] dataset, which consists of 900 upper jaw models and 900 lower jaw models. For each type of model, 600 models are used as the training set, and 300 for testing. The segmentation results demonstrate that TeethGNN [20] outperforms other methods, as shown in Table 2. To achieve optimal segmentation results, we further smooth the tooth boundaries through the graph-cut.

Effects of Encoder. Our TL-DETR network structure utilizes the TeethGNN [20] encoder to extract geometric features. We replace it with the encoders from DGCNN [13] and TSGCNet [18], and the quantitative results are shown in the second sub-table of Table 3.

Effects of Graph-Cut. Although TeethGNN [20] achieves high segmentation accuracy, isolated prediction errors and unsmooth tooth boundaries can negatively impact landmark localization. To address this, we applied graph-cut for further optimization. The comparison results with and without graph-cut are shown in the third sub-table of Table 3, demonstrating that the quality of the segmented tooth patches significantly affects the performance of subsequent landmark detection.

Effects of Decay Factor. The decay factor σ in the Gaussian kernel of Eq. 1 is a crucial parameter. A decay factor that is too small may result in insufficient prominence of the landmarks, while a factor that is too large can lead to sample imbalance. Therefore, we explored the results under various decay factors, as shown in the fourth sub-table of Table 3.

5 Limitation and Future Work

While TL-DETR demonstrates strong performance in detecting landmarks for individual teeth, there remains room for improvement. First, TL-DETR faces challenges when directly applied to jaw meshes. Second, our network architecture primarily relies on the TeethGNN backbone; exploring more effective mesh geometry feature extractors could improve accuracy, especially in landmark localization. Finally, TL-DETR's detection accuracy heavily depends on the quality of tooth patches derived from tooth segmentation. To further enhance landmark detection performance, we intend to investigate more robust and precise tooth segmentation methods.

6 Conclusion

This paper presents a two-stage 3D tooth landmark detection framework. In the first stage, jaw models are segmented through TeethGNN. In the second stage, TL-DETR is introduced to detect landmarks on the cropped tooth patches. TL-DETR features two branches: one for the fixed landmarks and the other for a variable number of cusp landmarks. The cusp landmark branch is trained using bipartite matching loss, enabling end-to-end inference. Extensive experiments show that our method not only eliminates the need for post-processing but also accurately predicts both fixed and a variable number of cusp landmarks, demonstrating excellent generalization and robustness.

References

1. Ben-Hamadou, A., et al.: Teeth3DS+: an extended benchmark for intraoral 3D scans analysis. arXiv preprint arXiv:2210.06094 (2022)
2. Ben-Hamadou, A., et al.: Teeth3DS: a benchmark for teeth segmentation and labeling from intra-oral 3D scans. arXiv preprint arXiv:2210.06094 (2022)
3. Ben-Hamadou, A., et al.: 3DTeethSeg'22: 3D teeth scan segmentation and labeling challenge. arXiv preprint arXiv:2305.18277 (2023)
4. Cui, Z.: TSegNet: An efficient and accurate tooth segmentation network on 3D dental model. Med. Image Anal. **69**, 101949 (2021)
5. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 209–216 (1997)
6. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: MeshCNN: a network with an edge. ACM Trans. Graph. **38**(4), 1–12 (2019)
7. Hu, S.M.: Subdivision-based mesh convolution networks. ACM Trans. Graph. **41**(3), 1–16 (2022)
8. Jin, H., Shen, Y., Lou, J., Zhou, K., Zheng, Y.: KeypointDETR: an end-to-end 3D keypoint detector. In: European Conference on Computer Vision (2024)
9. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)

10. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
11. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inform. Process. Syst. **30** (2017)
12. Wang, C., Wei, G., Wei, G., Wang, W., Zhou, Y.: Tooth alignment network based on landmark constraints and hierarchical graph structure. IEEE Trans. Vis. Comput. Graph. **30**(2), 1457–1469 (2022)
13. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**(5), 1–12 (2019)
14. Wei, G., et al.: Dense representative tooth landmark/axis detection network on 3D model. Comput. Aided Geom. Des. **94**, 102077 (2022)
15. Wei, G., Ma, L., Wang, C., Desrosiers, C., Zhou, Y.: Multi-task joint learning of 3D keypoint saliency and correspondence estimation. Comput. Aided Des. **141**, 103105 (2021)
16. Wu, T.H., Lian, C., Lee, S., et al.: Two-stage mesh deep learning for automated tooth segmentation and landmark localization on 3D intraoral scans. IEEE Trans. Med. Imaging **41**(11), 3158–3166 (2022)
17. You, Y., et al.: KeypointNet: a large-scale 3D keypoint dataset aggregated from numerous human annotations. In: IEEE Conference on Computer Vision Pattern Recognition, pp. 13647–13656 (2020)
18. Zhang, L., et al.: TSGCNet: discriminative geometric feature learning with two-stream graph convolutional network for 3D dental model segmentation. In: IEEE Conference on Computer Vision Pattern Recognition, pp. 6699–6708 (2021)
19. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259–16268 (2021)
20. Zheng, Y., Chen, B., Shen, Y., Shen, K.: TeethGNN: semantic 3D teeth segmentation with graph neural networks. IEEE Trans. Vis. Comput. Graph. **29**(7), 3158–3168 (2022)